

Sistema de Rastreabilidade de Peças com Estações de Processo

Objetivo Geral

Desenvolver um sistema de rastreabilidade de peças industriais, com controle de cadastro, movimentação entre estações, histórico completo e validação do fluxo de processo. O sistema deverá ser composto por:

- Uma API RESTful desenvolvida em .NET 6 ou superior.
- Uma interface web desenvolvida com Angular 15 ou superior.
- Controle de versionamento utilizando Git.

Regras de Negócio

1. Fluxo Obrigatório de Estações

Cada peça deve obrigatoriamente passar pelas seguintes três estações, na ordem definida:

- Estação 1: Recebimento
- Estação 2: Montagem
- Estação 3: Inspeção Final

2. Validação da Ordem de Processo

- A movimentação de uma peça só pode ocorrer para a próxima estação na ordem definida.
- Não é permitido pular ou retroceder estações.
- Após a passagem pela última estação, o status da peça deve ser atualizado para **Finalizada**.

3. Histórico de Movimentações

- Todas as movimentações devem ser registradas com data, origem, destino e responsável.
- O histórico deve ser consultável por peça.

4. Cadastro e Consulta

- As estações devem ser cadastradas previamente com sua ordem no processo.
- Permitir consultar as movimentações de uma peça específica pelo código.

- O Código da peça deve ser único e não deverá ser permitido o cadastro de 2 códigos iguais para diferentes peças.

Funcionalidades Requeridas

Back-End (.NET)

1. Implementar CRUD completo para as entidades **Peça** e **Estação**.
2. Implementar o registro de movimentações, com validação da ordem de processo.
3. Atualizar automaticamente o status da peça com base na estação atual.
4. Criar endpoint para consulta do histórico de movimentações por peça.
5. Implementar testes unitários com **xUnit**, cobrindo:
 - Validação da ordem das estações.
 - Registro de movimentações válidas e rejeição de inválidas.
 - Atualização do status da peça.

Observação:

Não é obrigatório utilizar banco de dados. O uso de armazenamento em memória é suficiente para fins desta atividade.

Entretanto, a utilização de banco de dados relacional (como SQL Server ou SQLite), com implementação de **migrations** via **Entity Framework**, será considerada um diferencial positivo.

Boas práticas recomendadas:

- Organização em camadas.
- Uso adequado de orientação a objetos.
- Separação de responsabilidades e reutilização de código.

Front-End (Angular)

1. Desenvolver componentes para:
 - Cadastro e listagem de peças.
 - Cadastro e listagem de estações.
 - Registro de movimentações.

- Consulta de histórico por peça.
2. Integrar com a API via serviços Angular.

Critérios de Avaliação

- Implementação correta das regras de negócio e validações.
- Cobertura adequada de testes unitários.
- Interface funcional, clara e responsiva.
- Código limpo, modular e bem documentado.
- Aplicação de boas práticas de desenvolvimento e arquitetura.

Instruções para Entrega

- O candidato terá **até o dia 31/07/2025** para realizar o desenvolvimento a partir do recebimento deste desafio.
- Ao iniciar o desenvolvimento, o candidato deve **criar um repositório no Git**.
- Ao finalizar, deve **enviar o link do repositório** para avaliação.